

#2

# BUNDESREPUBLIK DEUTSCHLAND

**PRIORITY DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)



REC'D 16 SEP 2003	
WIPO	PCT

## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 102 37 875.4

**Anmeldetag:** 19. August 2002

**Anmelder/Inhaber:** Siemens Aktiengesellschaft, München/DE

**Bezeichnung:** Vorrichtung, insbesondere Automatisierungsgerät,  
mit in Datei gespeicherter Dateiverzeichnisstruktur

**IPC:** G 06 F 17/30

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 13. Mai 2003  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

Wenner

Best Available Copy

## Beschreibung

Vorrichtung, insbesondere Automatisierungsgerät, mit in Datei gespeicherter Dateiverzeichnisstruktur

5

Die Erfindung betrifft eine Vorrichtung, insbesondere Automatisierungsgerät, mit in Datei gespeicherter Dateiverzeichnisstruktur.

- 10 Dateiverzeichnisstrukturen, insbesondere hierarchische Dateiverzeichnisstrukturen, dienen u.a. zum Ordnen bzw. schnellen Wiederfinden von einzelnen Dateien innerhalb einer großen Datenmenge, die sich beispielsweise auf einer Festplatte eines Rechners befindet. Ein typisches Beispiel hierfür ist z.B.
- 15 der Windows Explorer der Firma Microsoft.

- Des Weiteren halten Internet Technologien wie beispielsweise HTTP, die Fähigkeit, mit Internet Browsern, wie beispielsweise dem Microsoft Internet Explorer oder Netscape zu kommunizieren, immer mehr Einzug in die Welt der Automatisierungstechnik. Dies gilt im Besonderen für den Bereich der Embedded Systeme. Unter Embedded Systeme oder auch Embedded Devices versteht man Systeme mit eingebetteten Computerfunktionen, wie man sie beispielsweise in intelligenten Sensoren und Aktuatoren, Busbaugruppen sowie in vielen Geräte- und Maschinensteuerungen bzw. anderen Anwendungen der Automatisierungstechnik findet. Auch für den Embedded Systems-Bereich gibt es mittlerweile eine Reihe von Web-Servern, die für eine geeignete Kommunikation der Automatisierungsgeräte mit bzw. via
- 20
- 25
- 30 Internet/Intranet sorgen.

- Alle diese Web-Server für Embedded Systeme haben im Wesentlichen die gleichen Probleme. Einerseits bestehen nur geringe Ressourcen auf der Zielmaschine bzw. dem jeweiligen Automatisierungsgerät, wodurch in der Regel ein File-System bzw. eine Dateiverzeichnisstruktur fehlt. Da Objekte, die mit Internet Technologien adressiert werden sollen, über eine sog. URL
- 35

(Uniform Resource Locator) adressiert werden und diese URLs zur Adressierung eine hierarchische Struktur, wie sie von Filesystemen angeboten werden, benutzen, können Objekte, die sich auf solchen Embedded Systemen befinden, nicht ohne weiteres adressiert werden. Andererseits besteht aber die Notwendigkeit eines Fernzugriffs auf diese Automatisierungsgeräte bzw. Embedded Systeme, da diese in der Regel räumlich weit entfernt und unbeaufsichtigt laufen. Eine Fernbedienung bzw. Remote Konfiguration z.B. beim Auftreten von Fehlern, zu Wartungszwecken etc. kann die Anwesenheit von Personal vor Ort reduzieren und ist damit kostengünstig. Um diese Anforderung zu lösen, werden heutzutage File-Systeme aufwändig nachimplementiert, wobei Files, also Dateien, dann beispielsweise als unabhängige Speicherblöcke existieren, die beispielsweise über den so genannten FTP-Dienst (File Transfer Protocol) verwaltet werden müssen. Neben dem Problem, dass die Verwaltung solcher einzelner Speicherblöcke sehr aufwändig ist, muss ein solcher FTP-Server in der Regel zusätzlich auf dem Embedded System implementiert werden, was die Ressourcen auf der Zielmaschine weiter reduziert. Darüber hinaus sind Automatisierungssysteme, die durch Firewalls geschützt werden, nicht notwendigerweise für einen FTP-Server durchlässig, der Fernzugriff ist also unter Umständen verwehrt.

Aufgabe der vorliegenden Erfindung ist es, eine Vorrichtung anzugeben, die eine Implementierung einer Dateiverzeichnisstruktur vor allem auf Embedded Systemen bzw. Automatisierungsgeräten auch ohne eigene Dateiverzeichnisstruktur ermöglicht.

Diese Aufgabe wird durch eine Vorrichtung mit einem Speicher zur Speicherung einer Dateiverzeichnisstruktur mit wenigstens einer ersten und einer zweiten Hierarchieebene, welche als untergeordnete Ebene der ersten Hierarchieebene ausgebildet ist, mit wenigstens einem ersten Dateiverzeichnis, das sich auf der ersten Hierarchieebene befindet, mit wenigstens einem zweiten Dateiverzeichnis, das sich auf der zweiten Hierar-

chieebene befindet und mit wenigstens einer ersten Datei, die sich auf einer der zwei Hierarchieebenen oder einer darunter liegenden Hierarchieebene befindet, dadurch gelöst, dass die Dateiverzeichnisstruktur in einer zweiten Datei abgespeichert

5 ist, wobei die Dateiverzeichnisstruktur einen Teil des Inhalts oder den Gesamtinhalt der zweiten Datei darstellt, wobei jedes Dateiverzeichnis und jede Datei der Dateiverzeichnisstruktur in der zweiten Datei nacheinander aufgelistet, durch wenigstens ein charakteristisches Anfangssymbol

10 und/oder wenigstens ein charakteristisches Endesymbol gekennzeichnet und die Inhalte jedes Dateiverzeichnisses und jeder Datei der Dateiverzeichnisstruktur jeweils zwischen den zugeordneten beiden charakteristischen Symbolen abgelegt sind.

15 Diese Aufgabe wird durch ein Verfahren zur Abbildung einer hierarchischen ersten Dateiverzeichnisstruktur mit wenigstens einer ersten und einer zweiten Hierarchieebene, welche als untergeordnete Ebene der ersten Hierarchieebene ausgebildet ist, mit wenigstens einem ersten Dateiverzeichnis, das sich

20 auf der ersten Hierarchieebene befindet, mit wenigstens einem zweiten Dateiverzeichnis, das sich auf der zweiten Hierarchieebene befindet und mit wenigstens einer ersten Datei, die sich auf einer der zwei Hierarchieebenen oder einer darunter liegenden Hierarchieebene befindet, in eine zweite Datei ge-

25 löst, wobei die aus der hierarchischen ersten Dateiverzeichnisstruktur abgebildete zweite Dateiverzeichnisstruktur einen Teil des Inhalts oder den Gesamtinhalt der zweiten Datei darstellt, wobei jedes Dateiverzeichnis und jede Datei der abzubildenden hierarchischen Dateiverzeichnisstruktur in der

30 zweiten Datei nacheinander aufgelistet, durch wenigstens ein charakteristisches Anfangssymbol und/oder wenigstens ein charakteristisches Endesymbol gekennzeichnet und die Inhalte jedes Dateiverzeichnisses und jeder Datei der abzubildenden Dateiverzeichnisstruktur jeweils zwischen den zugeordneten bei-

35 den charakteristischen Symbolen abgelegt werden.

Nach einer bevorzugten Ausführungsform der Erfindung wird zur Beschreibung der Dateiverzeichnisstruktur eine internetfähige Sprache verwendet. Nach einer weiteren bevorzugten Ausführungsform der Erfindung ist die zweite Datei, in der die Dateiverzeichnisstruktur abgebildet ist, eine XML-Datei, wobei zur Beschreibung, insbesondere der Dateiverzeichnisstruktur, wird die Sprache XML verwendet. Selbstverständlich können auch andere geeignete, internetfähige Sprachen neben XML, wie beispielsweise HTML/XHTML, CSS, Java Script, CGI/Perl, etc. bei Bedarf verwendet werden. XML-Dateien können sehr leicht aufgebaut werden und haben nur einen geringen Speicherplatzbedarf. Sie können ebenfalls sehr leicht remote via Intra- bzw. Internet versendet werden.

Nach einer weiteren bevorzugten Ausführungsform wird sowohl für jedes charakteristische Anfangssymbol als auch für jedes charakteristische Endesymbol jeweils eine neue Zeile in der zweiten Datei verwendet. Dabei dient als jeweiliges charakteristisches Anfangssymbol beispielsweise die Bezeichnung des jeweiligen Dateiverzeichnisses bzw. der jeweiligen Datei und als jeweiliges charakteristisches Endesymbol die Bezeichnung des jeweiligen Dateiverzeichnisses bzw. der jeweiligen Datei, wobei zusätzlich ein vorgebbares Zeichen vorangestellt wird. Ein solches vorgebbares Zeichen ist beispielsweise das Zeichen „/“. Es könnte selbstverständlich auch ein beliebiges anderes Zeichen, Sonderzeichen, Zahl, etc. sein. Des Weiteren wird der Inhalt jedes Dateiverzeichnisses einer Hierarchieebene der abzubildenden Dateiverzeichnisstruktur einschließlich des Inhaltes der Dateiverzeichnisse und Dateien aller Unterhierarchieebenen dieses Dateiverzeichnisses zwischen dem charakteristischen Anfangssymbol und dem charakteristischen Endesymbol des betreffenden Dateiverzeichnisses abgelegt. Durch diese Klammerstruktur zwischen charakteristischem Anfangs- und Endesymbol für jedes Dateiverzeichnis bzw. jede Datei wird die Abbildung der Hierarchie einschließlich der Auflistung der jeweiligen Inhalte der Dateiverzeichnisse bzw. Dateien erreicht. Für XML wird dafür beispielsweise der so

genannte „Tag“-Mechanismus benutzt. Zusätzlich wird jeweils zwischen dem charakteristischen Anfangs- bzw. Endesymbol des jeweiligen Dateiverzeichnisses bzw. der jeweiligen Datei der Inhalt der entsprechenden Dateiverzeichnisse bzw. der jeweiligen Dateien abgelegt, wodurch ein hierarchisches Dateiverzeichnis einschließlich der jeweiligen Inhalte in einer solchen Datei, beispielsweise XML-Datei, nacheinander aufgelistet wird. Gleichzeitig ist es möglich, gezielt und schnell auf die entsprechenden Inhalte der Dateiverzeichnisse und Dateien zuzugreifen, da diese durch die jeweiligen charakteristischen Anfangs- bzw. Endesymbole direkt adressierbar sind.

Des weiteren stellt die abgebildete Datenverzeichnisstruktur nur einen Teil des Inhalts der zweiten Datei dar, wobei der Anfang dieses Teils durch wenigstens ein charakteristisches Anfangssymbol und der Ende dieses Teils durch wenigstens ein charakteristisches Endesymbol gekennzeichnet wird, wobei die abgebildete Dateiverzeichnisstruktur zwischen dem charakteristischen Anfangssymbol und dem charakteristischen Endesymbol abgelegt wird. Beispielsweise kann für das charakteristische Anfangssymbol für die abgebildete Dateiverzeichnisstruktur das Wort „BASE“ und als charakteristisches Endesymbol für die abgebildete Dateiverzeichnisstruktur das Wort „BASE“ und zusätzliches Voranstellen beispielsweise des vorgebbaren Zeichens „/“ gekennzeichnet werden. Damit hat man eine Trennung der reinen Dateiverzeichnisstruktur von eventuellen zusätzlichen Informationen erreicht, die man im Anschluss an die so gekennzeichnete Dateiverzeichnisstruktur in einer solchen XML-Datei unterbringen will.

Nach einer weiteren bevorzugten Ausführungsform der Erfindung sind in der zweiten Datei zusätzlich zu dem Teil, der die abgebildete Dateiverzeichnisstruktur enthält, weitere Teile mit anderen Inhalten vorhanden, die jeweils durch wenigstens ein charakteristisches Anfangssymbol und wenigstens ein charakteristisches Endesymbol gekennzeichnet bzw. abgetrennt sind. Beispielsweise können in wenigstens einem der weiteren ge-

kennzeichneten Teile der zweiten Datei Konfigurationsdaten bzw. in wenigstens einem der weiteren gekennzeichneten Teile der zweiten Datei Ergebnis- und/oder Fehlercodes abgelegt werden. Durch eine solche Klammerstruktur können in der zweiten Datei weitere charakteristische Daten für die Konfiguration der jeweiligen Embedded Systeme bzw. Automatisierungsgeräte und/oder Rückmeldungen der jeweiligen Embedded Systeme bzw. Automatisierungsgeräte, insbesondere Ergebnis- und/oder Fehlercodes, in separaten geklammerten Teilen der zweiten Datei abgelegt werden, die jeweils durch ein entsprechendes charakteristisches Anfangssymbol und wenigstens ein charakteristisches Endesymbol gekennzeichnet bzw. abgetrennt sind.

Dabei wird die zweite Datei vorzugsweise in einem Gerät eingesetzt, das keine eigene Dateiverzeichnisstruktur aufweist. Dadurch, dass die zweite Datei eine abgebildete Dateiverzeichnisstruktur enthält, ersetzt diese die sonst übliche eigene Dateiverzeichnisstruktur des Geräts.

Von besonderem Vorteil ist es darüber hinaus, dass die Vorrichtung Mittel zum Empfang und/oder zur Speicherung der zweiten Datei über ein Kommunikationsnetz, insbesondere Inter- und/oder Intranet und/oder eine Funkverbindung, aufweist. Durch ein solches ferngesteuertes Laden, beispielsweise einer XML-Datei über Inter- und/oder Intranet und/oder beispielsweise drahtlos über eine Funkanbindung, in ein Automatisierungsgerät ist es möglich, auch räumlich sehr weit entfernt liegende Automatisierungsgeräte auch ohne persönliches Erscheinen von Technikern bzw. anderem Personal vor Ort anzusprechen.

Ein weitere überaus vorteilhafte Weiterbildung der Erfindung ist es, dass nach dem Laden der zweiten Datei in die Vorrichtung automatisch eine Konfiguration der Vorrichtung mit den in der zweiten Datei vorhandenen Konfigurationsdaten durchführbar ist. Dadurch können Automatisierungsgeräte, die räumlich sehr weit entfernt liegen, sehr leicht automatisch kon-

figuriert werden, was den Inbetriebsetzungsaufwand und die damit verbundenen Kosten erheblich reduziert.

5 Weiterhin ist es sehr vorteilhaft, dass nach dem Laden der zweiten Datei in die Vorrichtung, die Vorrichtung als Web-Server einsetzbar ist. Dadurch ist es möglich, über das Internet bzw. Intranet das Embedded Device, bzw. Automatisierungsgerät direkt remote anzusprechen, wodurch ebenfalls ein  
10 Erscheinen von menschlichem Personal vor Ort vermieden werden kann.

Eine überaus vorteilhafte Weiterbildung der Erfindung ist, dass eine Aktualisierung der abgebildeten Dateiverzeichnisstruktur durch Überschreiben der ursprünglichen Dateiversion  
15 der zweiten Datei mit einer neuen Dateiversion durchführbar ist. Darüber hinaus ist durch das Überschreiben der ursprünglichen Dateiversion der zweiten Datei mit einer neuen Dateiversion eine Aktualisierung der Konfigurationsdaten durchführbar. Es wird dabei also automatisch eine Aktualisierung  
20 der Konfigurationsdaten durchgeführt, wodurch aktuelle Daten sehr schnell auf das Embedded System bzw. das Automatisierungsgerät übertragen werden können. Das Überschreiben der ursprünglichen Dateiversion der zweiten Datei mit einer neuen Dateiversion kann dabei selbstverständlich ferngesteuert über  
25 Inter- und/oder Intranet und/oder drahtlos über eine entsprechende Funkanbindung durchgeführt werden.

Nach einer weiteren bevorzugten Ausführungsform der Erfindung sind nach der Aktualisierung der zweiten Datei die bisher  
30 eingestellten Konfigurationsdaten der Vorrichtung, in die die ursprüngliche Dateiversion der zweiten Datei geladen war, automatisch überprüfbar und ggf. anpassbar. Dadurch ist es möglich, sehr leicht und sehr schnell auf aktuelle Veränderungen, die ein Anpassen der Konfigurationsdaten des Automatisierungsgeräts und die Konfiguration des Geräts selbst nötig  
35 erscheinen lassen, zu reagieren.



Überaus vorteilhaft ist es darüber hinaus, wenn die Vorrichtung ein Embedded Device und/oder ein Automatisierungsgerät ist und diese Vorrichtung in einem Automatisierungssystem eingesetzt wird.

5

Von besonderem Vorteil ist es darüber hinaus, dass das offenbarte Verfahren in Automatisierungssystemen, insbesondere bei und in Verpackungsmaschinen, Pressen, Kunststoffspritzmaschinen, Textilmaschinen, Druckmaschinen, Werkzeugmaschinen, Roboter, Handlingsystemen, Holzverarbeitungs-  
10 maschinen, Glasverarbeitungs-  
maschinen, Keramikverarbeitungs-  
maschinen sowie Hebezeugen eingesetzt bzw. verwendet werden kann.

15

Mit dem offenbarten Verfahren können also beispielsweise HTML-Seiten, andere Dateien, sowie Konfigurationsdaten, etc. die für ein Automatisierungsgerät, bzw. Embedded Device benötigt werden, beispielsweise in Form einer XML-Seite in ein Automatisierungsgerät geladen und dort gespeichert werden, wodurch ein solches Gerät auch als Web-Server eingesetzt werden kann. Es ist darüber hinaus ebenfalls möglich, binäre Objekte, z.B. in den HTML-Seiten gelinkte Grafiken entweder mit Hilfe der so genannten BASE 64 Encoding-Methode innerhalb der XML-Seite zu speichern oder separat davon. In diesem Fall  
20 sind die binären Objekte per Link aus der XML-Seite zugreifbar.  
25

30

Dadurch, dass die benötigte Dateiverzeichnisstruktur incl. Dateiverzeichnissen und Dateien, beispielsweise als HTML-Seiten, samt Inhalt innerhalb einer einzigen XML-Seite aufgelistet wird, ist keine aufwändige Verwaltung dieser einzelnen Objekte notwendig. Darüber hinaus werden zur Bearbeitung einer solchen XML-Datei nur Standard Tools benötigt. Die Verwendung proprietärer Tools, bei denen die Verfügbarkeit in der Zukunft nicht unbedingt gewährleistet ist, ist dadurch  
35 überflüssig.

Darüber hinaus ist es, wie oben offenbart, sehr leicht möglich, eine einfache Konfiguration eines solchen als Web-Server arbeitenden Automatisierungsgeräts durch internetfähige Mittel, wie z.B. HTTP-Befehle, durchzuführen. Darüber hinaus werden keine weiteren Dienste benötigt. Durch den Remote Zugriff ist eine schnelle und flexible Erweiterung und/oder Änderung von beispielsweise Konfigurationsdaten möglich, eine Abwärtskompatibilität ist dadurch automatisch gegeben. Mit dem offenbarten Verfahren ist es zusätzlich sehr leicht möglich, eine solche XML-Seite, auch Web Site genannt, lokal auf einem Rechner, z.B. mit dem Tool Front Page, zu erstellen, als XML-Datei abzuspeichern, sie über das Inter- und/oder Intranet an das Embedded Device zu senden und dort durch erstmaliges Implementieren oder durch Überschreiben der ursprünglichen XML-Datei den Inhalt der XML-Datei zu übernehmen und alle notwendigen Aktualisierungen durchzuführen. Man spricht in diesem Fall auch von einem Publishing Mechanismus.

Die Notwendigkeit einer hierarchischen Dateiverzeichnisstruktur, wie sie jeder Web-Server benötigt und bisher in anderer Form abgebildet wurde, wird durch die Nutzung der hierarchischen Strukturmittel der XML-Sprache, den sogenannten „Tags“, mit oben genannten Vorteilen realisiert und befriedigt.

Im Weiteren werden bevorzugte Ausführungsbeispiele der Erfindung mit Bezugnahme auf die Zeichnungen näher erläutert. Es zeigen:

FIG 1 eine Prinzipdarstellung einer klassischen hierarchischen Dateiverzeichnisstruktur,

FIG 2 eine Prinzipdarstellung einer XML-Datei mit hierarchischer Dateiverzeichnisstruktur,

FIG 3 ein Flussdiagramm zur Implementierung einer XML-Datei und

FIG 4 eine erfindungsgemäße Vorrichtung mit Speicher und XML-Datei in einem Automatisierungssystem.

FIG 1 zeigt die prinzipielle Darstellung einer klassischen hierarchischen Dateiverzeichnisstruktur 35 auch File-System genannt. Eine solche Dateiverzeichnisstruktur 35 besteht aus einem Root-Verzeichnis, welches die oberste Hierarchieebene 1 einer solchen Dateiverzeichnisstruktur 35 darstellt. Alle weiteren Dateiunterverzeichnisse bzw. Dateien sind hierarchisch dem Root-Verzeichnis, also der ersten Hierarchieebene 1 untergeordnet. In der FIG 1 sind Dateiverzeichnisse als Rechtecke dargestellt. Dateien sind als Ovale dargestellt, wobei die Verbindungslinien zwischen den Rechtecken bzw. Ovalen die Zugehörigkeit zu Objekten bzw. zu Hierarchieebenen kennzeichnen. Die Punkte, die die Verbindungslinien unterbrechen, kennzeichnen die Möglichkeit von beliebigen Unterverzeichnisstrukturen. Unter beliebigen Unterverzeichnisstrukturen sind sowohl Dateiverzeichnisse, auch Directories genannt, als auch Dateien, auch Files genannt, zu verstehen.

Im gezeigten Beispiel besteht die zweite Hierarchieebene 2 aus beispielsweise drei Directories, von denen der Übersichtlichkeit halber nur das Directory 6 Disc\_C bezeichnet wurde. Bei diesen Directories handelt es sich beispielsweise um die Darstellung von verschiedenen Partitionen einer Festplatte eines beliebigen Rechners. Selbstverständlich können damit beispielsweise auch Laufwerke, also externe Speichermedien bzw. Peripheriegeräte des betreffenden Rechners angesprochen werden, beispielsweise Diskettenlaufwerke, in denen Disketten oder CD-ROM-Laufwerke, in denen CD-ROMs gelesen werden können.

In der Figur 1 ist beispielsweise das Directory 6, also beispielsweise die Partition Disc\_C einer entsprechenden Festplatte, in mehrere Unter-Directories, auch Sub Directories genannt, unterteilt, die sich auf der dritten Hierarchieebene 3 befinden. Auch hier ist der Übersichtlichkeit halber nur

das Sub Directory 7 DIR\_C2 bezeichnet worden. Wie aus der Figur 1 ebenfalls ersichtlich ist, können sich auch auf dieser Ebene einzelne Dateien befinden. Aus Gründen der Übersichtlichkeit ist beispielhaft nur die Datei 10 C-File1.html bezeichnet worden. Selbstverständlich können sich beliebig viele weitere Dateien auf der dritten und auf jeder anderen Hierarchieebene befinden. Dies ist jeweils durch Punkte angedeutet.

- 10 In der Figur 1 ist beispielhaft das Directory 7 Dir\_C2 weiter in Unter-Directories unterteilt, die die vierte Hierarchiestufe 4 darstellen. Aus Gründen der Übersichtlichkeit ist nur das Sub Directory 8, bzw. Unter-Directory, SubDirC2\_b bezeichnet worden. Bei der vierten Hierarchiestufe 4 ist schließlich eine beispielhafte Unterteilung des Sub-Directories 8 SubDirC2\_b vorgenommen worden, auf der sich in diesem Beispiel nur Dateien befinden, von denen aus Gründen der Übersichtlichkeit nur die Datei 9 C2\_b-file1.html bezeichnet worden ist. Diese bilden in dem vorliegenden Beispiel die fünfte Hierarchiestufe 5.

- Selbstverständlich können bei einem beliebigen File-System unabhängig von dem gezeigten Beispiel beliebige weitere Unterverzeichnisstrukturen vorhanden sein. Selbst die einzelnen Hierarchiestufen können beliebige Verzeichnisse bzw. Files aufweisen. Auch die Namen der einzelnen Directories bzw. Dateiverzeichnisse sowie der Dateien bzw. Files sind hier nur beispielhaft gewählt worden. Sie können mit beliebigen anderen Namen bezeichnet werden, die beispielsweise aus Zahlen und/oder Buchstaben und/oder Sonderzeichen gebildet werden können. Somit bilden Objekte einer niedrigeren Hierarchiestufe den Inhalt des entsprechenden zugehörigen Dateiverzeichnisses auf der höheren Hierarchiestufe. Die Zugehörigkeit wird durch die entsprechenden Linien deutlich. So kennzeichnen beispielsweise alle Objekte der dritten Hierarchiestufe 3 im gezeigten Beispiel den Inhalt des Directories 6 Disc\_C der zweiten Hierarchiestufe 2. Andere Objekte der Hierarchiestufe

2 wurden in diesem Beispiel nicht verfeinert. Analog dazu ist dies bei den Hierarchiestufen 3, 4 bzw. 5 dargestellt. Aus Gründen der Übersichtlichkeit wurde darauf verzichtet, den Inhalt einer Datei selbst, beispielsweise der Datei 9 C2\_B-  
5 File1.html, explizit darzustellen.

Die Figur 2 zeigt eine erfindungsgemäß erstellte Datei 11 mit abgebildeter hierarchischer Dateiverzeichnisstruktur 12. Die Datei 11 wird im folgenden auch als zweite Datei oder XML-  
10 Datei bezeichnet. Die verwendeten Begriffe in der Figur 2 sind dabei an die Begriffe der Figur 1 angelehnt. Selbstverständlich ist auch die Darstellung einer Dateiverzeichnisstruktur 12 möglich, ohne dass eine physikalische Vorlage einer Dateiverzeichnisstruktur 35, wie sie in der FIG. 1 dargestellt ist, existiert. Die dargestellte zweite Datei 11 ist  
15 beispielsweise eine XML-Datei und beinhaltet beispielhaft drei große Hauptblöcke, die abgebildete Dateiverzeichnisstruktur 12, den Abschnitt 13, der mit SERVER\_CONFIGURATION bezeichnet ist und beispielsweise Konfigurationsdaten für ein  
20 Automatisierungsgerät bzw. einen Web-Server enthält, sowie den Abschnitt 14, der mit RESULT\_CODES bezeichnet worden ist und in dem beispielsweise Ergebnis- und/oder Fehlercodes, insbesondere vom Automatisierungsgerät bzw. dem Web-Server abgelegt und dann über remote-Zugriff ausgewertet werden können. Selbstverständlich können weitere Abschnitte in der Datei 11 im Anschluss an die Abschnitte 12, 13, bzw. 14 abgelegt werden, in denen weitere benötigte Daten hinterlegt werden können. Die Punkte vor dem Abschnitt 12 bzw. nach dem Abschnitt 14 verdeutlichen, dass noch weitere XML-Daten, Zeilen  
25 bzw. Abschnitte in der Datei 11 eingefügt werden können. In der Datei 11 ist, wie vorher bereits erwähnt, eine hierarchische Dateiverzeichnisstruktur 12 beispielhaft dargestellt.

Kennzeichen der XML-Sprache und vieler internetfähiger Sprachen, ist, dass jede Zeile mit dem Zeichen „<“ beginnt, danach der eigentliche Inhalt der Zeile folgt und die jeweilige Zeile mit dem Zeichen „>“ beendet wird. Jedes der in der XML-  
35

Datei 11 beispielhaft dargestellten Abschnitte 12, 13, 14, beginnt mit einem charakteristischen Anfangssymbol und endet mit einem charakteristischen Endesymbol. Der Inhalt jedes Abschnittes ist zwischen diesen beiden Symbolen eingeklammert.

5 Das jeweilige charakteristische Anfangssymbol sowie das jeweilige charakteristische Endesymbol erhält jeweils eine neue Zeile innerhalb der XML-Datei. Aufgrund einer übersichtlichen Darstellung sind lediglich das charakteristische Anfangssymbol 15 sowie das charakteristische Endesymbol 16 der hierarchischen Dateiverzeichnisstruktur 12 entsprechend bezeichnet worden. Das charakteristische Anfangssymbol 15 der abgebildeten Dateiverzeichnisstruktur 12 hat beispielsweise die Bezeichnung „BASE“. Um das Ende eines entsprechenden Abschnitts in einer XML-Datei kennzeichnen zu können, wird der Bezeichnung, die das charakteristische Anfangssymbol 15 des betreffenden Abschnitts bildet, beispielsweise das vorgebbare Zeichen 36 „/“ vorangestellt. Im Prinzip könnte jedoch auch ein beliebiges anderes Zeichen vorgegeben werden. Im vorliegenden Beispiel der abgebildeten hierarchischen Dateiverzeichnisstruktur 12 ist somit das charakteristische Endesymbol 16 der Begriff „BASE“, dem das Zeichen „/“ vorangestellt wird. Die gesamte abzubildende Dateiverzeichnisstruktur 12 sowie die Inhalte der jeweiligen Directories bzw. Sub-Directories bzw. Dateien werden dann zwischen dem charakteristischen Anfangssymbol 15 und dem charakteristischen Endesymbol 16 des Abschnitts 12 aufgelistet und dargestellt.

So ist der Abschnitt in der gezeigten Datei 11, der die abgebildete Dateiverzeichnisstruktur 12 beispielhaft kennzeichnet, aus verschiedenen Unterabschnitten 17, 18, 19, 20 sowie 21 aufgebaut. So kennzeichnet der hier gezeigte Unterabschnitt 17 ein Verzeichnis mit dem Namen Disc\_A. Die Punkte im Unterabschnitt 17 zwischen dem charakteristischen Anfangssymbol, das durch die Bezeichnung Disc\_A gekennzeichnet ist, und dem charakteristischen Endesymbol, das durch die Bezeichnung /Disc\_A gekennzeichnet ist, deuten an, dass sich hier eine beliebig verschachtelte Unterverzeichnis- bzw. Datei-

struktur befinden kann, auf dessen Darstellung aus Gründen der Übersichtlichkeit jedoch verzichtet wurde. Das charakteristische Anfangssymbol 15 der Dateiverzeichnisstruktur 12 symbolisiert die oberste Hierarchieebene 1 der Dateiverzeichnisstruktur 35 aus Figur 1, entspricht also dem Root-Verzeichnis. Das charakteristische Anfangssymbol des Unterabschnitts 17 mit der Bezeichnung Disc\_A symbolisiert den Beginn des ersten Objekts der zweiten Hierarchieebene unterhalb des Root-Verzeichnisses.

Das charakteristische Anfangssymbol des Unterabschnitts 18, mit der Bezeichnung Disc\_C symbolisiert den Beginn eines weiteren Objekts der zweiten Hierarchieebene unterhalb des Root-Verzeichnisses, also der obersten Hierarchieebene. Unterabschnitte 17 und 18 kennzeichnen also Objekte, die auf derselben Hierarchieebene unterhalb des Root-Verzeichnisses stehen.

Aus Gründen der Übersichtlichkeit wurde im Unterabschnitt 18 beispielhaft nur ein Unterverzeichnis weiter verzweigt. Es handelt sich dabei um den Unterabschnitt 19. Der Unterabschnitt 19 kennzeichnet das Unterverzeichnis mit der Bezeichnung Dir\_C2, das mit dem charakteristischen Anfangssymbol mit derselben Bezeichnung beginnt und mit dem charakteristischen Endesymbol /Dir\_C2 beendet wird. Dazwischen befindet sich wiederum die komplette Unterverzeichnisstruktur bzw. die enthaltenen Dateien sowie die Inhalte der Dateien, die sich im Unterverzeichnis Dir\_C2 befinden. Der Unterabschnitt 19 entspricht somit einem Objekt der dritten Hierarchieebene (vergleiche FIG 1).

Zur Verdeutlichung der möglichen weiteren Verzweigungen wurden beispielhaft noch die Unterabschnitte 20 und 21 innerhalb des Unterabschnittes 19 der Dateiverzeichnisstruktur 12 in der XML-Datei 11 aufgelistet. Dabei beginnt der Unterabschnitt 20, der ein Unterverzeichnis des Verzeichnisses Dir\_C2 kennzeichnet, mit dem charakteristischen Anfangssymbol SubDirC2\_b und endet mit dem charakteristischen Endesymbol

/SubDirC2\_b. Dazwischen ist wiederum der Inhalt des so bezeichneten Unterverzeichnisses enthalten. Der Inhalt des Unterverzeichnisses 20 ist in diesem Beispiel der Unterabschnitt 21, der kein weiteres Unterverzeichnis aufweist, sondern eine einzelne Datei mit der Bezeichnung C2\_b-File1.html darstellt, womit gleichzeitig das charakteristische Anfangssymbol bezeichnet ist. Das Ende der Datei wird wiederum mit dem charakteristischen Endesymbol /C2\_b-File1.html bezeichnet. Die Punkte zwischen dem charakteristischen Anfangssymbol bzw. dem charakteristischen Endesymbol des Unterabschnitts 21 sollen den Inhalt der Datei andeuten, auf dessen Wiedergabe jedoch aus Gründen der Übersichtlichkeit verzichtet wird. Unterabschnitt 20 entspricht somit einem Objekt der vierten Hierarchieebene 4 aus der Figur 1 und Unterabschnitt 21 entspricht somit einem Objekt der fünften Hierarchieebene 5, ebenfalls aus der Figur 1.

Durch die Verwendung einer internetfähigen Sprache, am Beispiel gezeigt durch die Verwendung der Sprache XML, wird die Darstellung einer Dateiverzeichnisstruktur 12 und/oder Abbildung einer beliebigen hierarchischen Dateiverzeichnisstruktur 35 in eine Dateiverzeichnisstruktur 12 innerhalb einer Datei 11 ermöglicht. Die Versendung einer solchen Datei 11 über Internet und/oder Intranet oder auch über eine Funkanbindung sowie der Empfang einer solchen Datei 11 von einem Embedded Device, insbesondere Automatisierungsgerät, ist sehr einfach und problemlos möglich. Zur problemlosen Identifizierung der einzelnen Unterabschnitte 17, 18, 19, 20 bzw. 21 der Dateiverzeichnisstruktur 12 wird vorteilhafterweise als charakteristisches Anfangssymbol jedes Unterabschnittes jeweils die Bezeichnung des betreffenden Dateiverzeichnisses bzw. der abzubildenden Datei verwendet. Um den Anfang bzw. das Ende eines beliebigen Dateiverzeichnisses bzw. einer beliebigen Datei schnell und problemlos identifizieren zu können, ist es darüber hinaus vorteilhaft, zumindest das charakteristische Anfangssymbol und das jeweilige charakteristische Endesymbol jeweils in einer neuen Zeile aufzulisten. Der Inhalt der so



abzubildenden Dateiverzeichnisse bzw. Dateien befindet sich dann vorteilhafterweise geklammert zwischen dem jeweiligen charakteristischen Anfangssymbol und Endesymbol der jeweiligen Objekte. Die abgebildete Dateiverzeichnisstruktur 12 bildet vorteilhafterweise nur einen Teil der XML-Datei 11, welche zusätzlich noch weitere Abschnitte, insbesondere Abschnitte 13 und 14, wie oben erwähnt, enthält, darüber hinaus aber noch weitere Abschnitte enthalten kann. Die anderen Abschnitte sind ebenfalls vorteilhafterweise zur direkten Adressierung über Internet bzw. Intranet durch ein charakteristisches Anfangssymbol gekennzeichnet, und das Ende ist jeweils wiederum durch ein charakteristisches Endesymbol gekennzeichnet. So kann beispielsweise das Unterverzeichnis Dir\_C2, das gleichzeitig das charakteristische Anfangssymbol des Unterabschnittes 19 aus der Dateiverzeichnisstruktur 12 der XML-Datei 11 kennzeichnet, mittels einer URL-Adresse direkt angesprochen werden. Eine solche URL-Adresse hat z.B. die Form „http://Serveradresse/BASE/Disc\_C/Dir\_C2/SubDirC2\_b/“. Mit dem Begriff „Serveradresse“ ist die Bezeichnung bzw. Adresse gemeint, mittels derer das betreffende Automatisierungsgerät, das auch als Web-Server arbeitet, über Internet und/oder Intranet angesprochen und/oder adressiert werden kann. Mit der oben erwähnten vollständigen URL-Adresse kann die entsprechende Html-Seite, bzw. Datei, in diesem Beispiel die Seite C2\_b-File1.html direkt remote über Inter- bzw. Intranet angesprochen und aufgerufen werden. Der Inhalt der betreffenden Seite, die sich in der XML-Datei 11 befindet, kann somit auf einem räumlich entfernten PC dargestellt und dessen Inhalt gegebenenfalls sehr leicht verändert und/oder ausgewertet, gespeichert, überschrieben, etc. werden.

FIG 3 zeigt ein Flussdiagramm zur Implementierung einer XML-Datei mit einer Dateiverzeichnisstruktur 12 auf einer erfindungsgemäßen Vorrichtung, insbesondere einem Automatisierungsgerät. Mit dem Implementieren einer solchen Datei 11 wird darüber hinaus die Web-Server-Funktionalität auf das

betreffende Embedded Device transportiert. Vorteilhafterweise wird eine solche XML-Datei 11 auf ein Embedded Device, insbesondere Automatisierungsgerät, gebracht, welches standardmäßig keine eigene Dateiverzeichnisstruktur aufweist. So wird

5 im Schritt 22 eine XML-Datei 11 beispielsweise lokal auf einem beliebigen Rechner erstellt und mit den notwendigen Abschnitten, beispielsweise der Dateiverzeichnisstruktur 12 (vgl. Figur 2), sowie mit anderen Abschnitten, die Daten

10 beinhalten, welche für die Funktionalität eines Web-Servers notwendig sind, beispielsweise Abschnitt 13 der XML-Datei 11, der die entsprechenden Konfigurationsdaten für einen solchen Web-Server enthält, oder auch weiteren Abschnitten für beliebige weitere notwendige Daten ausgestattet. Sobald die beispielhafte Datei 11 fertiggestellt ist, wird sie im Schritt

15 23 via ein angeschlossenes Kommunikationsnetz, beispielsweise Inter- bzw. Intranet, oder auch per geeigneter Funkanbindung, an ein an dieses Kommunikationsnetz angeschlossenes Embedded Device gesendet.

20 Bei oder nach dem Empfang der Datei 11 durch das Automatisierungsgerät wird im Schritt 24 kontrolliert, ob auf dem Embedded Device bereits eine Datei 11 vorhanden ist. Falls noch keine Datei 11 vorhanden ist, wird die gesendete Datei 11 im Schritt 25 auf dem entsprechenden Embedded Device bzw. Automatisierungsgerät implementiert und danach im Schritt 26 das Embedded Device mit den im beispielsweise Abschnitt 14 verfügbaren Konfigurationsdaten der Datei 11 automatisch konfiguriert.

30 Ist auf dem entsprechenden Automatisierungsgerät bereits eine Datei 11 vorhanden, so wird die ursprüngliche Datei 11 im Schritt 27 von der neuen, also aktuellen, Datei 11 überschrieben, die beispielsweise andere, aktuelle Konfigurationsdaten enthalten kann. Insbesondere kann die neue aktuelle

35 Dateiversion der Datei 11 bei Bedarf aufgrund geänderter Anforderungen, bzw. Randbedingungen auch eine neue, geänderte Dateiverzeichnisstruktur aufweisen. Im Schritt 28 wird danach

kontrolliert, ob die Konfiguration des entsprechenden Embedded Devices bzw. des entsprechenden Automatisierungsgeräts aktualisiert werden muss. Dies kann dann der Fall sein, wenn mit der neuen XML-Datei aufgrund aktueller Veränderungen beispielsweise neue Konfigurationsdaten an das Automatisierungs-  
5 gerät gesendet wurden. Auch diese Aktualisierung erfolgt vorteilhafterweise automatisch nach der Überprüfung, ob sich die entsprechenden Konfigurationsdaten der Zielmaschine geändert haben, um auf einen Einsatz von Wartungs- und/oder sonstigem  
10 Personal vor Ort verzichten zu können.

Im Schritt 29 kann das entsprechende Embedded Device bzw. das entsprechende Automatisierungsgerät die Funktion als Web-  
Server aufnehmen und in Betrieb gehen. Eine direkte Kommuni-  
15 kation über ein Inter- bzw. Intranet von einem räumlich entfernten Rechner, also eine remote-Verbindung, wird somit sehr leicht gewährleistet, was sich kostengünstig auf Wartungs- bzw. Bedienungsaufwand auswirkt. Ein solches Embedded Device kann vorteilhafterweise in einem beliebigen Automatisierungs-  
20 system eingesetzt werden.

Fig. 4 zeigt eine erfindungsgemäßes Automatisierungsgerät 30 mit Speicher 31 und XML-Datei 11 in einem Automatisierungssystem. Das Automatisierungsgerät 30 stellt dabei beispielsweise eine SPS oder sonstige Steuerung beispielsweise für einen Motor 39, einen intelligenten Sensor oder ein beliebiges anderes Automatisierungsgerät dar. Es ist mit einem Speicher 31 ausgestattet, in dem eine Datei 11, insbesondere XML-  
Datei, gespeichert werden kann. Die Datei 11 kann direkt über  
30 eine eventuelle Schnittstelle in den Speicher 31 geladen werden. Bevorzugt wird die Datei 11 jedoch auf einem lokalen Rechner 34 erstellt, von dem sie mittels einer ersten Kommunikationsverbindung 32, über ein Kommunikationsnetz 33, insbesondere Internet und/oder Intranet und/oder eine geeignete  
35 Funkanbindung und mittels einer zweiten Kommunikationsverbindung 37, mittels derer das Automatisierungsgerät 30 an das Kommunikationsnetz 33 angeschlossen ist, an das Automatisie-

rungsgerät 30 gesendet wird. Das Automatisierungsgerät 30 weist dabei Mittel zum Empfang der Datei 11 auf und speichert diese in dem Speicher 31 ab und kann dann als Web-Server arbeiten. Das Automatisierungsgerät 30 ist dann von einem beliebigen anderen Teilnehmer 38 des Kommunikationssystems, insbesondere Automatisierungssystem, direkt adressierbar. Auf die Daten die in den einzelnen Abschnitten der Datei 11 aufgelistet sind, kann somit remote zugegriffen werden. Diese Daten können also von einem beliebigen, anderen Teilnehmer 38 des Kommunikationssystems schnell und flexibel beispielsweise verändert, angepasst, ausgelesen, ausgewertet, etc. werden, ohne dass eine Anwesenheit von entsprechendem Personal zu Bedienungs- und/oder Wartungszwecken vor Ort nötig ist.

15 Zusammengefasst handelt es sich bei der vorliegenden Erfindung um eine Vorrichtung und ein Verfahren, das dazu dient, um auf Automatisierungsgeräten, so genannten Embedded Devices, die in der Regel keine eigene Dateiverzeichnisstruktur 12 aufweisen, sehr leicht mit Hilfe einer internetfähigen Sprache, insbesondere XML, eine Dateiverzeichnisstruktur 12 innerhalb einer solchen XML-Datei 11 darzustellen und/oder abzubilden und durch Senden der Datei 11 auf das Embedded Device zu bringen, so dass die Zielmaschine als Web-Server arbeitet und so einen remote-Zugriff ermöglicht.

## Patentansprüche

1. Vorrichtung (30) mit einem Speicher (31) zur Speicherung einer Dateiverzeichnisstruktur (12) mit wenigstens einer ersten (1) und einer zweiten (2) Hierarchieebene, welche als untergeordnete Ebene der ersten Hierarchieebene (1) ausgebildet ist, mit wenigstens einem ersten Dateiverzeichnis, das sich auf der ersten Hierarchieebene (1) befindet, mit wenigstens einem zweiten Dateiverzeichnis (6), das sich auf der zweiten Hierarchieebene (2) befindet und mit wenigstens einer ersten Datei (9), die sich auf einer der zwei Hierarchieebenen (1, 2) oder einer darunter liegenden Hierarchieebene (3,4,5) befindet,
- d a d u r c h        g e k e n n z e i c h n e t,
- 15 dass die Dateiverzeichnisstruktur (12) in einer zweiten Datei (11) abgespeichert ist, wobei die Dateiverzeichnisstruktur (12) einen Teil des Inhalts oder den Gesamtinhalt der zweiten Datei (11) darstellt, wobei jedes Dateiverzeichnis und jede Datei der Dateiverzeichnisstruktur (12) in der zweiten Datei
- 20 (11) nacheinander aufgelistet, durch wenigstens ein charakteristisches Anfangssymbol (15) und/oder wenigstens ein charakteristisches Endesymbol (16) gekennzeichnet und die Inhalte jedes Dateiverzeichnisses und jeder Datei der Dateiverzeichnisstruktur (12) jeweils zwischen den zugeordneten beiden
- 25 charakteristischen Symbolen (15,16) abgelegt sind.
2. Vorrichtung nach Anspruch 1,
- d a d u r c h        g e k e n n z e i c h n e t,
- dass zur Beschreibung der Dateiverzeichnisstruktur (12) eine
- 30 internetfähige Sprache verwendet wird.
3. Vorrichtung nach einem der Ansprüche 1 oder 2,
- d a d u r c h        g e k e n n z e i c h n e t,
- dass die zweite Datei (11), in der die Dateiverzeichnisstruktur (12) abgespeichert ist, eine XML-Datei ist und zur Beschreibung, insbesondere der Dateiverzeichnisstruktur (12),
- 35 die Sprache XML verwendet wird.

4. Vorrichtung nach einem der vorherigen Ansprüche,  
dadurch gekennzeichnet,  
dass sowohl für jedes charakteristische Anfangssymbol (15)  
als auch für jedes charakteristische Endesymbol (16) jeweils  
5 eine neue Zeile in der zweiten Datei (11) verwendet wird.

5. Vorrichtung nach einem der vorherigen Ansprüche,  
dadurch gekennzeichnet,  
dass als jeweiliges charakteristisches Anfangssymbol (15) die  
10 Bezeichnung des jeweiligen Dateiverzeichnisses bzw. der je-  
weiligen Datei verwendet wird, und dass als jeweiliges cha-  
rakteristisches Endesymbol (16) die Bezeichnung des jeweili-  
gen Dateiverzeichnisses bzw. der jeweiligen Datei verwendet  
wird und zusätzlich ein vorgebbares Zeichen vorangestellt  
15 wird.

6. Vorrichtung nach einem der vorherigen Ansprüche,  
dadurch gekennzeichnet,  
dass in der zweiten Datei (11) zusätzlich zu dem Abschnitt,  
20 der die Dateiverzeichnisstruktur (12) enthält, weitere Ab-  
schnitte (13, 14) mit anderen Inhalten vorhanden sind, die  
jeweils durch wenigstens ein charakteristisches Anfangssymbol  
(15) und wenigstens ein charakteristisches Endesymbol (16)  
gekennzeichnet bzw. abgetrennt sind.

25 7. Vorrichtung nach Anspruch 6,  
dadurch gekennzeichnet,  
dass in wenigstens einem der weiteren gekennzeichneten Ab-  
schnitte (13, 14) der zweiten Datei (11) Konfigurationsdaten  
30 abgelegt sind.

8. Vorrichtung nach Anspruch 6,  
dadurch gekennzeichnet,  
dass in wenigstens einem der weiteren gekennzeichneten Ab-  
35 schnitte (13, 14) der zweiten Datei (11) Ergebnis- und/oder  
Fehlercodes abgelegt sind.

9. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
dass die Vorrichtung Mittel zum Empfang und/oder zur Speicherung der zweiten Datei (11) über ein Kommunikationsnetz, insbesondere Inter- und/oder Intranet (33) und/oder eine Funk-  
5 verbindung, aufweist.
10. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
10 dass nach dem Laden der zweiten Datei (11) in die Vorrichtung automatisch eine Konfiguration der Vorrichtung mit den in der zweiten Datei (11) vorhandenen Konfigurationsdaten durchführbar ist.
- 15 11. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
dass nach dem Laden der zweiten Datei (11) in die Vorrichtung, die Vorrichtung als Webserver einsetzbar ist.
- 20 12. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
dass eine Aktualisierung der Dateiverzeichnisstruktur (12) durch Überschreiben der ursprünglichen Dateiversion der zweiten Datei (11) mit einer neuen Dateiversion durchführbar ist.
- 25 13. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
dass durch das Überschreiben der ursprünglichen Dateiversion der zweiten Datei (11) mit einer neuen Dateiversion eine Aktualisierung der Konfigurationsdaten durchführbar ist.  
30
14. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h g e k e n n z e i c h n e t,  
dass nach der Aktualisierung der zweiten Datei (11) die bisher eingestellten Konfigurationsdaten der Vorrichtung, in das  
35 die ursprüngliche Dateiversion der zweiten Datei (11) geladen war, automatisch überprüfbar und ggf. anpassbar sind.

15. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h        g e k e n n z e i c h n e t,  
dass die Vorrichtung ein Embedded Device ist.

5

16. Vorrichtung nach einem der vorherigen Ansprüche,  
d a d u r c h        g e k e n n z e i c h n e t,  
dass die Vorrichtung ein Automatisierungsgerät ist.

10 17. Automatisierungssystem mit wenigstens einer Vorrichtung  
nach einem der Ansprüche 1 bis 16.

15 18. Verfahren zur Abbildung einer hierarchischen ersten Da-  
teiverzeichnisstruktur (35) mit wenigstens einer ersten (1)  
und einer zweiten (2) Hierarchieebene, welche als untergeord-  
nete Ebene der ersten Hierarchieebene (1) ausgebildet ist,  
mit wenigstens einem ersten Dateiverzeichnis, das sich auf  
der ersten Hierarchieebene (1) befindet, mit wenigstens einem  
20 zweiten Dateiverzeichnis (6), das sich auf der zweiten Hie-  
rarchieebene (2) befindet und mit wenigstens einer ersten Da-  
tei (9), die sich auf einer der zwei Hierarchieebenen (1, 2)  
oder einer darunter liegenden Hierarchieebene (3,4,5) befin-  
det, in eine zweite Datei (11), wobei die aus der hierarchi-  
schen ersten Dateiverzeichnisstruktur (35) abgebildete zweite  
25 Dateiverzeichnisstruktur (12) einen Teil des Inhalts oder den  
Gesamtinhalt der zweiten Datei (11) darstellt, wobei jedes  
Dateiverzeichnis und jede Datei der abzubildenden hierarchi-  
schen Dateiverzeichnisstruktur (35) in der zweiten Datei (11)  
nacheinander aufgelistet, durch wenigstens ein charakteristi-  
30 sches Anfangssymbol (15) und/oder wenigstens ein charakteris-  
tisches Endesymbol (16) gekennzeichnet und die Inhalte jedes  
Dateiverzeichnisses und jeder Datei der abzubildenden Datei-  
verzeichnisstruktur (35) jeweils zwischen den zugeordneten  
beiden charakteristischen Symbolen (15,16) abgelegt werden.

35



## Zusammenfassung

Vorrichtung, insbesondere Automatisierungsgerät, mit in Datei gespeicherter Dateiverzeichnisstruktur

5

Bei der vorliegenden Erfindung handelt es sich um eine Vorrichtung und ein Verfahren, das dazu dient, um auf Automatisierungsgeräten, so genannten Embedded Devices, die in der Regel keine eigene Dateiverzeichnisstruktur (12) aufweisen, sehr leicht mit Hilfe einer internetfähigen Sprache, insbesondere XML, eine Dateiverzeichnisstruktur (12) innerhalb einer solchen XML-Datei (11) darzustellen und/oder abzubilden und durch Senden der Datei (11) auf das Embedded Device zu bringen, so dass die Zielmaschine als Web-Server arbeitet und so einen remote-Zugriff ermöglicht.

10

15

FIG 4

FIG 1

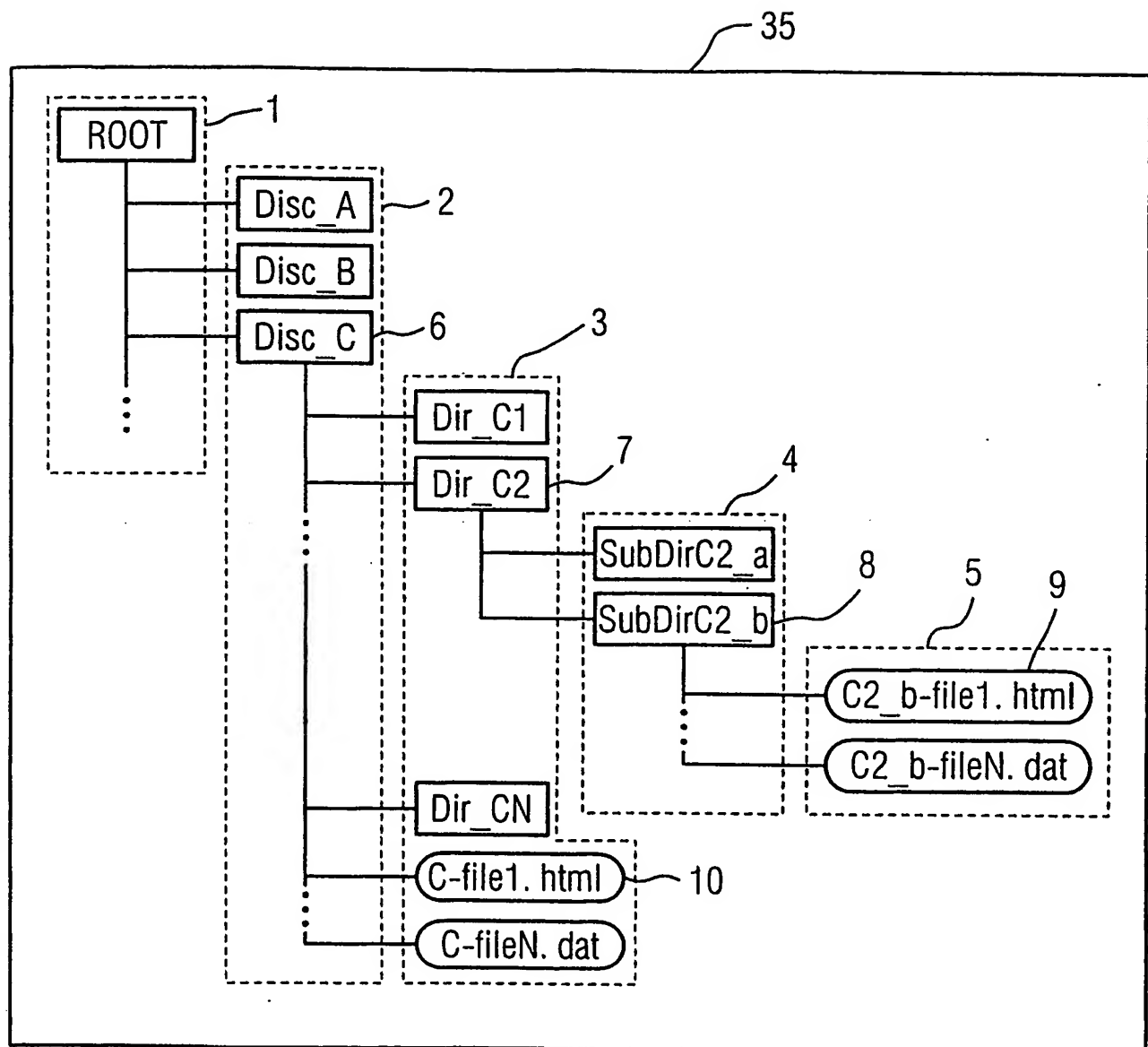


FIG 2

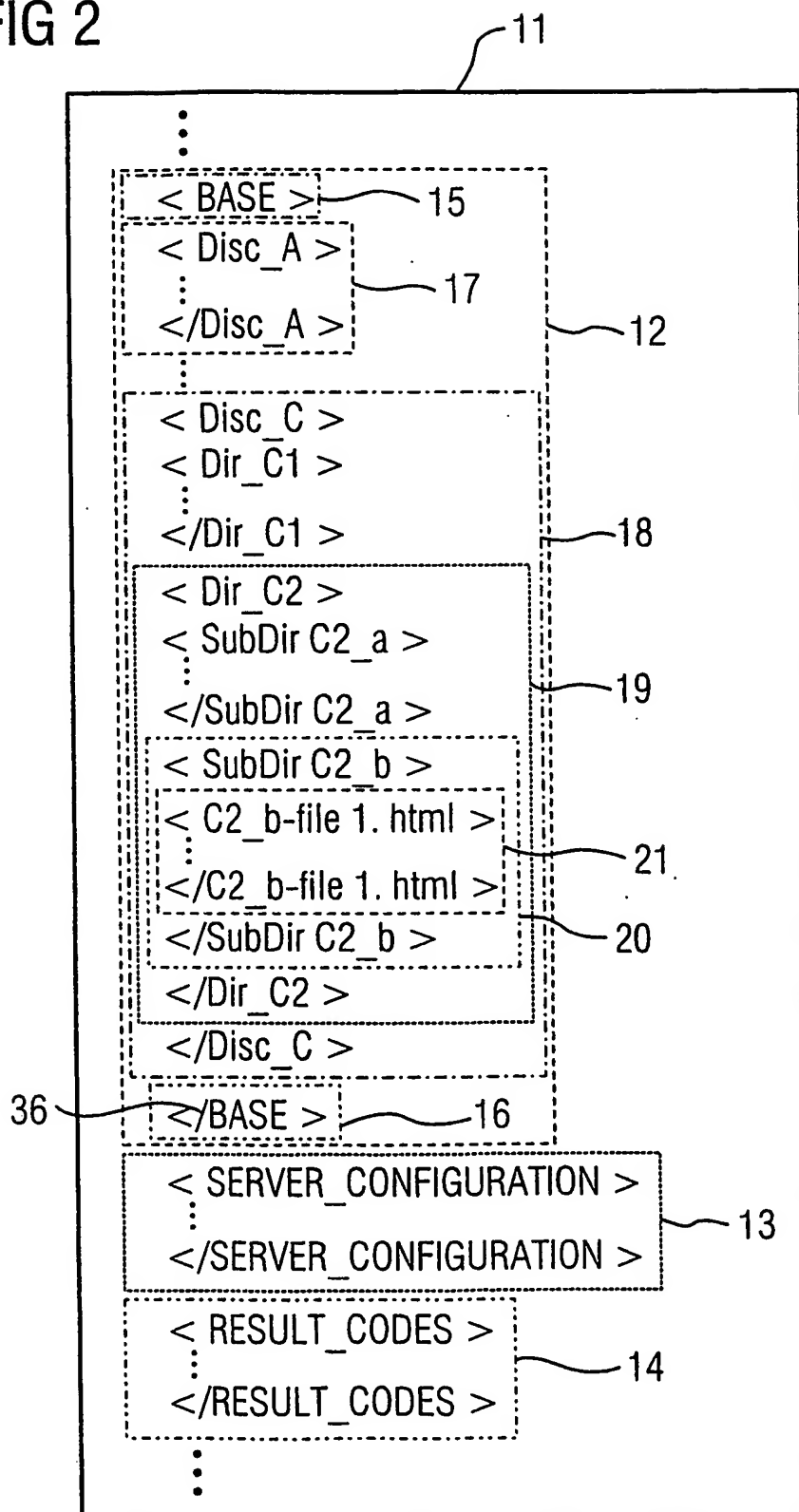


FIG 3

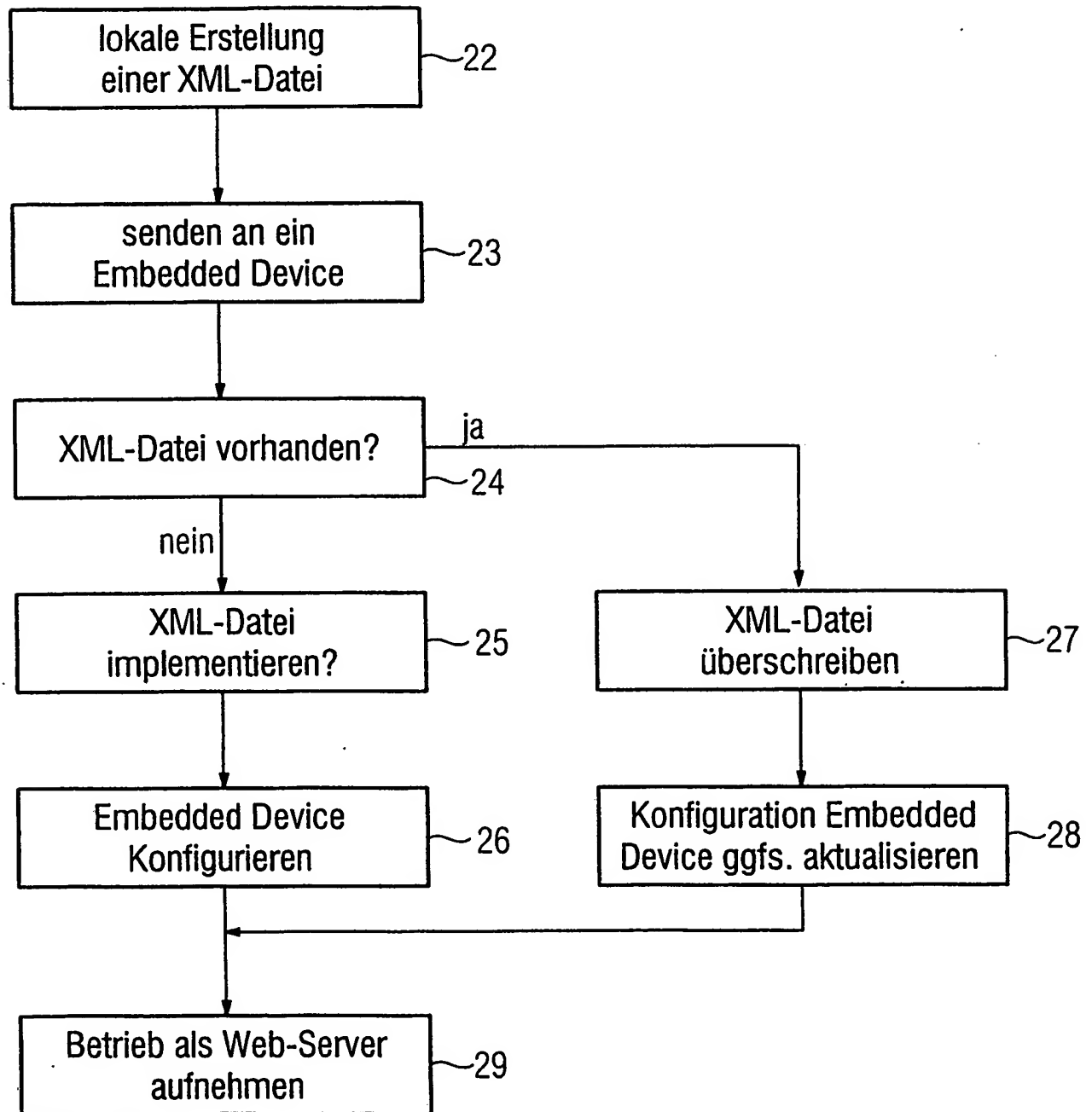
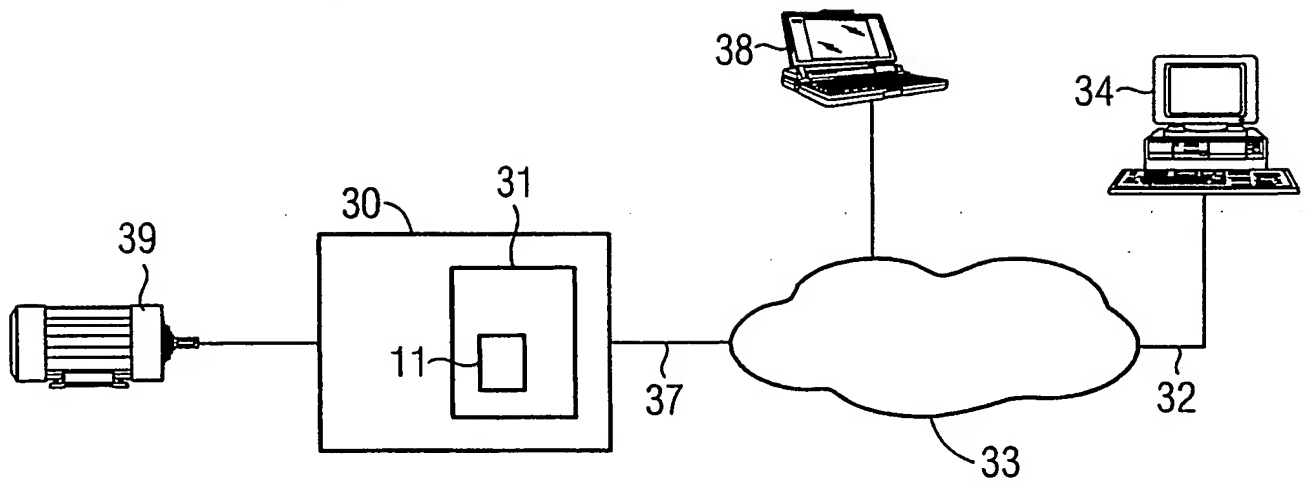


FIG 4



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**